REMARKS/ARGUMENTS

Claims 2-55 are pending in this application and are rejected under 35 U.S.C. §103. For at least the reasons set forth below, Applicants assert that all claims are in condition for allowance.

Rejection Under 35 U.S.C. § 103

Claims 2-55 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Dillingham (US 6,327,608) in view of Wolf et al. (US 5,818,447). As set forth in more detail below, the references fail to teach or suggest all the claim limitations as required by MPEP § 2143. Therefore, the rejection is unsupported by the art, and Applicants respectfully request that the rejection be withdrawn.

(a) *The References Fail to Teach or Suggest an Offline Action and Processing the Offline Action by a UI Server*

Independent claim 8 recites, "receiving a command from said client device, said command being indicative of an offline action performed by said client device; and said UI server processing said command for execution by said server-based application." The cited references fail to teach or suggest these limitations.

Examiner cites to Dillingham at Col. 2, lines 27-64 in support of anticipation of this limitation. However, this passage fails to teach or suggest the receipt of a command indicative of an offline action as claimed. Specifically, Dillingham discloses a file administration application that provides remote network access to a server-based file system via a client computer. The Dillingham system employs a client PC connected to the file server via the Internet. In this regard, one of the requirements of the Dillingham system is the PC-to-server connection. Indeed, Dillingham states that "[p]rior to gaining access to the server file system, the client first establishes a secure connection with the server." Col. 6, lines 31-32. Dillingham also states that "[t]he remote file administration architecture operates within the context of this secured environment. Accordingly, the commands described below...are all securely exchanged over the Internet..." Col. 6, lines 48-52. This description does not teach or suggest receiving an offline command that is processed by a UI server for execution by a server-based application as claimed.

In response to these arguments, Examiner argued:

> [T]he Offline Action can be done at a cache memory at the client PC. The action does not need to require a connection to the server. In Dillingham's system, the list of new files (which are obtained from the server) is cached locally. The data is cached on the client, it may be sorted or items may be selected without requiring another round trip to the server (column 8, lines 15-21).

OA dated 11/30/2005, p. 13. Applicants note that, even if this description teaches or suggests "receiving a command from said client device, said command being indicative of an offline action performed by said client device," the description provides no teaching or suggestion for "said UI server processing said command for execution by said server-based application." Specifically, Examiner's argument emphasizes that the "action does not need to require a connection to the server." It is unclear to applicants how, then, a UI server processes an offline command performed by a client device if the server and the client do not have a connection.

Furthermore, in Examiner's example it would be nonsensical for a server to process the offline action for execution by the server-based application as claimed if the offline action (i.e., sorting or selecting the data items, as recited by Examiner) was already performed on the client "without requiring another round trip to the server." If the sorting or selection was performed on the client of Dillingham as suggested by Examiner, it is unclear why the server would then process or execute the same sorting or selection actions. Moreover, there is no suggestion or motivation—nor does Examiner cite any suggestion or motivation—to modify Dillingham to execute a "sort" or "select" action on both the client and the server. Nowhere does the Dillingham reference, nor the other art of record, teach or suggest modifying the Dillingham reference to include this limitation.

Accordingly, the rejection does not establish a *prima facie* case of obviousness as defined by MPEP § 2143, and Applicants respectfully request reconsideration and withdrawal of this rejection.

(b) *<u>The References Fail to Teach or Suggest Retrieving or Generating a UI Form Definition that Corresponds to a Client Device's Platform or Capabilities</u>*

Claims 8, 20, 37, and 44 recite retrieving at a UI server or generating a UI form definition that <u>corresponds to or is based on a particular client device's platform or capabilities</u>. The rejection asserts that these limitations are taught or suggested by the Dillingham reference. Applicants respectfully submit that neither Dillingham nor Wolf, nor the combination thereof, teaches or suggests this limitation.

The UI server recited in independent claims 8, 20, 37, and 40 stores a UI form definition or identifier corresponding to a client device's platform or capabilities. The same form definition is also cached at the client device and is then populated with content from the UI server, i.e. source data items. The *Dillingham* reference discloses a distinct configuration and operation.

In contrast, the Dillingham reference discloses a distinct configuration and operation that does <u>not</u> provide a UI form definition that corresponds to or is based on a particular client device's platform or capabilities. Dillingham describes a typical Web server transmitting the <u>same</u> HTML and XML instructions to clients, <u>regardless of the clients' platform or capabilities</u>. Col. 3, line 62-Col. 4, line 4. Nowhere does Dillingham teach or suggest storing a UI form definition or identifier that corresponds to a particular client device as claimed. Nor does the reference teach or suggest selecting a web page from a plurality of web pages corresponding to a particular client's platform or capabilities as recited in amended claims 8, 37, and 44. The reference describes creating a "custom client-side object to cache," col. 9, lines 11-21, but this object is created in response to a <u>user</u> selection, col. 7, lines 32-36, col. 7, line 66-col. 8, line 13, not <u>client platforms or capabilities</u>.

In response to these arguments, Examiner argued:

> Dillingham shows the correspondence at column 2, lines 51-67 by citing "In one implementation, a server-side script creates a client-side script, which instantiates a custom client-side object to cache the returned directory data and to present that data in a dialog UI. Absent this process, the client-side browser would not know what files, folders and/or directories it will cache in order to present them in response to the specified query...The server returns the

> client-side executable and the directory data to the client. The client subsequently executes the script to instantiate a local object for caching the directory data."

OA dated 11/30/2005, pp. 12-13. This cited passage from Dillingham describes a "custom client-side object." However, this object is used merely to cache directory data, not a UI form definition, which is presented in a UI. Nowhere does this passage—nor any other portions of the reference—teach or suggest retrieving or generating a UI form definition that corresponds to or is based on a particular client device's platform or capabilities.

Additionally, Wolf fails to teach or suggest modifying Dillingham to achieve these limitations. The Wolf reference describes a system and method for editing e-mail messages with a full-featured word processor application that is separate from the email client. Wolf specifically describes embedded objects that can be edited when opened and displayed in a separate window with the UI provided by the application program that created the object. Col. 9, lines 28-39; see also col. 9, lines 39-54 (describing native and foreign frames). However, this description does not teach or suggest a UI server storing and a client caching the same UI form definition corresponding to a client device's platform or capabilities.

For this additional reason, the rejection does not establish a *prima facie* case of obviousness as defined by MPEP § 2143, and Applicants respectfully request reconsideration and withdrawal of the rejection.

(c) *The References Fail to Teach or Suggest Selecting UI Form Definition From a Plurality of UI Form Definitions*

Claims 8, 37, and 44 recite that the UI form definition is selected from a plurality of UI form definitions corresponding to a plurality of client platforms or client device capabilities. For substantially the same reasons asserted in section (b) above, the references fail to teach or suggest this limitation. Although the rejection asserts that this limitation is described in Dillingham, not only does Dillingham fail to teach or suggest a UI server storing and a client caching the same UI form definition corresponding to a client device's platform or capabilities, but nowhere does either reference describe selecting a UI form definition from a plurality of UI form definitions corresponding to a plurality of client platforms or client device capabilities.

For this additional reason, the rejection fails to establish a *prima facie* case of obviousness as defined by MPEP § 2143, and Applicants respectfully request reconsideration of the rejection.

This application now stands in allowable form and reconsideration and allowance are respectfully requested.

Respectfully submitted,

DORSEY & WHITNEY LLP
Customer Number 25763

Date: May 30, 2006

By: Christoph R Hilb
Christopher R. Hilberg,
Reg. No. 48,740
(612) 492-6694